# Deep Learning Practical Work 2-a
# Transfer Learning through feature extraction from a CNN

Caterina Leonelli & Luisa Neubauer

December 29, 2023

## 1 Introduction

This document contains a list of questions and their corresponding answers.

## 2 Questions and Answers

1. **Question 1:** Number of parameters - Knowing that the fully-connected layers account for the majority of the parameters in a model, give an estimate on the number of parameters of VGG16 (using the sizes given in Figure 1)

   - *Answer:* The number of parameters for this neural network will be dominated by the # parameters of the fully connected head. We will therefore say $N_{total}^{NN} = N_{CNN} + N_{FC} \approx N_{FC}$.
     There are three fully connected layers with input $\Rightarrow$ output sizes $25088 \rightarrow 4096 \rightarrow 4096 \rightarrow 1000$ $N_{FC} = (25088 * 4096) + (4096 * 4096) + (4096 * 1000) \approx 124$ Mio
     Counting all network parameters in pytorch yields around 138 Mio. So we were already quite close by assuming most of the parameters come from the fully connected part.

2. **Question 2:** Output size last layer - What is the output size of the last layer of VGG16? What does it correspond to?

   - *Answer:* The output size in this classification task after the last layer of the VGG is of size [1, #classes]. This gives us the class scores after the softmax function. Those are the probabilities per class and add up to one. Output values = probabilities indicating that the input image belongs to one of the 1000 classes.

3. **Question 3:** Bonus - Apply the network on several images of your choice and comment on the results

- *Answer:*

Figure 1: Input image

Output:

```
class id tensor(698)
class "palace"
probability of max class tensor(0.5005, grad_fn=<MaxBackward1>)
```

Explanation:

The VGG classification suggests that the object in the image is likely to depict a 'palace,' (class id of 698) with a probability of 50.05%. This is already quite a good categorization because the network has not been trained to recognize this specific type of building and a human would probably have gone with the same category. It categorizes with a probability/confidence level of more than 50%. This example demonstrated the network's ability to generalize from training images to a wide range of diverse testing data even though the network was not specifically trained to recognize this type of building. We can see that the features extracted are relatively robust.

4. **Question 4:** (Bonus) - Visualize several activation maps obtained after the first convolutional layer. How can we interpret them?

   - *Answer:* The activation maps (see figure 2) represent the features that the NN has learned after the first layer. Each (filter) map highlights pixels that are being activated by a particular filter (64 of them). These correspond to image features for example edges, specific colors or textures. Some filters show a higher degree of activation, which is indicative of these features not being present in the

input image. We can use these maps to understand to a certain degree, what features are important to the network and to get a certain degree of explainability.

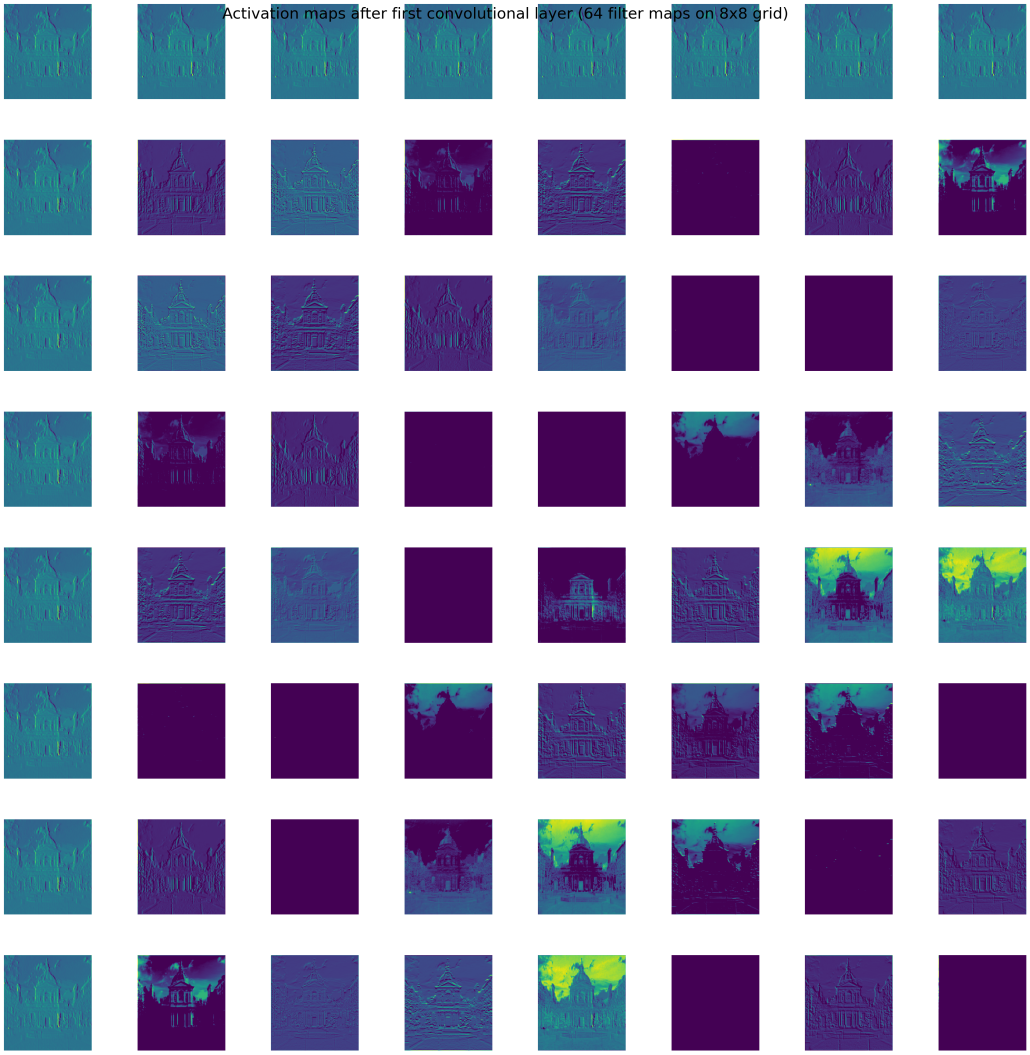


Figure 2: Activation maps after the first convolutional layer

5. **Question 5:** Why not train directly VGG16 on 15 Scene?

   - *Answer:* We could opt for a pretrained version - in this case VGG16 - for example because of limited computational resources. In general

traning a deep neural network with many millions or more paramters can require a very large number of CPU hours/processing power/memory/time or access to specialized hardware, which is very costly. Secondly, the 15Scene Dataset in miniscule in comparison to the ImageNet dataset. We might encounter overfitting if we train VGG on 15Scenes only.

6. **Question 6:** How can pre-training on ImageNet help classification for 15 Scene?

   - *Answer:* The images from 15 Scence were not drawn from the same data distribution $p_{data}(x)$ as ImageNet images. However, by pretraining on ImageNet, we assume that certain image properties are shared accross the two datasets. It might be more useful to use pretrained weights instead of a random initialization. Also, we achieve better generalization thanks to ImageNet.

7. **Question 7:** What limits can you see with feature extraction?

   - *Answer:* The features derived from the neural network are not by default interpretable and they might be tuned too much to a specific task. Also, in the case where the model architecture is not well-suited to our task, they features might be lacking.

8. **Question 8:** What is the impact of the layer at which the features are extracted?

   - *Answer:* Because we are working with a CNN, each feature that we extract at a certain network depth can only be a function of the inputs in the visual field of the respective, truncated CNN. In the limit of keeping only the first couple of layers, depending on the kernel sizes and strides, we can only extract very local features. Add to this that the features become highly non-linear with increased network depth and therefore, we can potentially keep fewer but more abstract and informative features.

9. **Question 9:** The images from 15 Scene are black and white, but VGG16 requires RGB images. How can we get around this problem?

   - *Answer:* As very often in ML, there is no one solution for every problem. In this case, the simplest solution would be to duplicate the grey scale channel across all three channels. For a more fancier solution, we could use another NN to colorize the images to then feed into the VGG16.

10. **Question 10:** Rather than training an independent classifier, is it possible to just use the neural network? Explain.

- *Answer:* Yes, this is possible. One way to go about using a NN as the classifier is to attach another NN at the end. We can then train this head using by (1) freezing the weights in the pre-trained network and only letting the model adjust the weights in the newly attached output NN or (2) use two different learning rates for the two parts of the NN. Many more strategies exist.

  If we don't want to train another network or a separate classifier to do the classification on our new dataset (15 Scene), we should at least assure, that the classes we are interested in, were part or the original classification labels (here: ImageNet). We can then ignore all the entries in the logits vector (1000 dimensional) that are not present in 15 Scenes and calculate the softmax on the remaining features.

11. **Question 11:** For every improvement that you test, explain your reasoning and comment on the obtained results.

    - *Answer:* To improve the result of the pre-trained classifier, we explored different options:

      (a) Hyper-parameter tuning for SVM on a grid using cross-validation: We provide regularization values C ranging from 0.001 to 1000 to the linear SVM and by GridSearch and 5-fold cross-validation pick the one that yields the highest accuracy. The resulting SVM uses $C = 0.01$ and an testing accuracy of 88.7%.

      (b) Attached NN head and retraining with frozen core network weights: We added a single fully connected layer to the output of the core NN (4096 features $\rightarrow$ 1000 categories). This perceptron is then trained on the training features and we employ early stopping as a regularization technique. With this simple setup, we reach a test accuracy of 88.4%. With a deeper MLP and more advanced training techniques, we are certain that one could boost the performance of the NN even more but this would go beyond the scope of this question.

      (c) Dimensionality reduction: PCA provided us a glimpse into how informative each single feature was and how many features one would need in order to reconstruct the data at a chosen threshold of total explained variance. Please consult figure 3 for the exact repartition of the variance per pca dimension.

    Please consult the notebook for the implementation, graphs and more details converning Question 11.

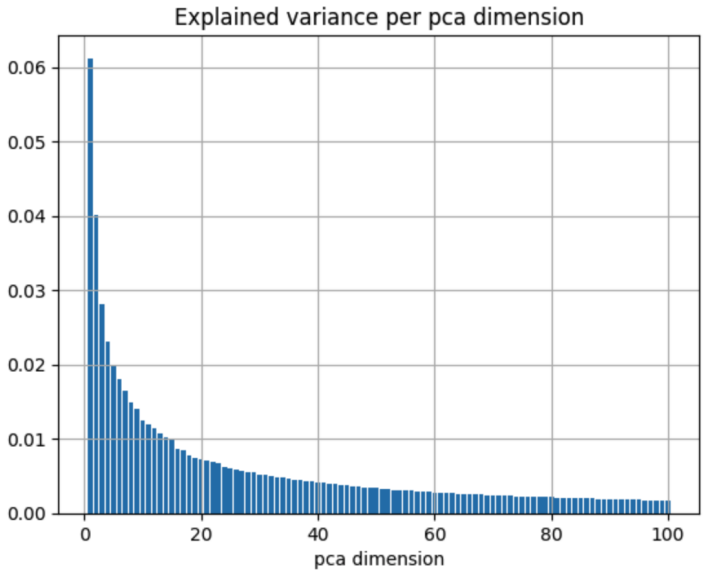


Figure 3: Result of PCA - Explained variance per PCA dimension